RESEARCH　　　　　　　　　　　　　　　　　　　　　　　　**Open Access**

Quantized Network Coding for correlated sources

Mahdy Nabaee[*] and Fabrice Labeau

Abstract

In this paper, we present a data gathering technique for sensor networks that exploits correlation between sensor data at different locations in the network. Contrary to distributed source coding, our method does not rely on knowledge of the source correlation model in each node although this knowledge is required at the decoder node. Similar to network coding, our proposed method (which we call Quantized Network Coding) propagates mixtures of packets through the network. The main conceptual difference between our technique and other existing methods is that Quantized Network Coding operates on the field of real numbers and not on a finite field. By exploiting principles borrowed from compressed sensing, we show that the proposed technique can achieve a good approximation of the network data at the sink node with only a few packets received and that this approximation gets progressively better as the number of received packets increases. We explain in the paper the theoretical foundation for the algorithm based on an analysis of the restricted isometry property of the corresponding measurement matrices. Extensive simulations comparing the proposed Quantized Network Coding to classic network coding and packet forwarding scenarios demonstrate its delay/distortion advantage.

Keywords: Linear network coding; Distributed source coding; Compressed sensing; Restricted isometry property; ℓ_1 minimization

1 Introduction

Flexible, low cost, and long-lasting implementation of wireless sensor networks has made them an unavoidable alternative for conventional wired sensing structures in a wide variety of applications, including medicine, transportation, and military [1]. As a relatively new technology, more challenges are faced in the networking aspects of communication than in the aspects of classic physical [2]. One of the introduced challenges is the gathering of sensed data at a central node of the network, where delivery delay, precision, and robustness to network changes are emerging issues.

Packet forwarding via routing is widely used in different implementations of sensor networks. While it achieves capacity rates in the case of multiple session unicast in lossless networks [3], packet forwarding requires an appropriate routing [4] protocol to be run. However, packet forwarding can lead to difficulties because of

its slow adaptation to the network changes, caused by deploying new node(s) or link failure(s).

Further, in the case of lossy networks, network coding offers a better error correction capability than packet forwarding, as a result of network diversity. Network coding [3] has been proposed as an alternative for packet forwarding in sensor networks [5,6]. Specifically, network coding sends a function of incoming packets to the intermediate nodes, as opposed to sending their original content. Furthermore, the usage of random linear functions, also known as random linear network coding, is proved to be sufficient in lossless networks [7,8]. Moreover, theoretical analysis shows that when network coding is used for transmission, no queuing is required to achieve the capacity rates of the network [3]. Network coding in lossy networks can result in improved achieved rate regions, compared to packet forwarding [9,10].

In the case of correlated sources, distributed source coding [11,12] on top of packet forwarding is proved to be sufficient, when dealing with networks of lossless links [13]. Similar to packet forwarding, network coding can be separately applied on top of distributed source coding

*Correspondence: mahdy.nabaee@mail.mcgill.ca
Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada

for correlated sources [14,15]. However, one has to perform joint source network decoding in order to achieve theoretical performance limits, which may not be feasible because of its computational complexity [15]. Different solutions have been proposed to tackle this practicality issue [16-18], by using low-density codes and sum product algorithm [19] for decoding.

Distributed source coding requires the availability of appropriate marginal coding rates at each encoder node; similarly, the deployment of joint source network decoding requires some knowledge of the correlation model of the sources on the encoding side. The assumption of this knowledge might not be practical in all cases, even more so when the source characteristics change over time.

Motivated by this observation, we aim to develop a data gathering and transmission scheme that, similar to network coding, does not rely on routing but at the same time can intrinsically take advantage of the source correlation. Our approach models source correlation through a sparsity or compressibility assumption; combined with a specific data gathering scheme inspired by network coding but acting in the real field, this assumption allows us to develop recovery algorithms at the sink node, which allow approximate data recovery with low delay. Our recovery mechanism will be based on ideas borrowed from compressed sensing [20,21] in which the inter-node correlation model of the messages, interpreted as near-sparsity in some domain, is used.

Recently, the idea of using compressed sensing and sparse recovery concepts in sensor networks has drawn a lot of attention [22-25]. Specifically, with the aid of the compressed sensing concepts, compression of inter-node correlated data without using their correlation model is done in [22,23]. Morevoer, in [26,27], theoretical discussion on sparse recovery of graph constrained measurements with an interest in network monitoring application is presented. Joint source, channel, and network coding was also proposed in [28], where random linear mixing was proposed for compression of temporally and spatially correlated sources. In [29], practical possibility of finite field network coding of highly correlated sources was investigated, with the aid of low-density codes and belief propagation-based decoding. Unfortunately, a solid theoretical investigation on the feasibility of adopting sparse recovery in random linear network coding has not been done previously.

Real network coding has shown interesting advantages over the conventional finite field network coding [30]. In our earlier work [31], we combined the idea of using real field network coding with the concepts of compressed sensing and proposed a non-adaptive distributed compression scheme, called *Quantized Network Coding* (QNC), for exactly sparse sources. Furthermore, in [32], we initiated a discussion on the theoretical

feasibility of compressed sensing-based network coding, using restricted isometry property of random matrices. In this paper, we extend our previous work from [31,32] in two specific ways: (i) we extend the network source model used from exactly sparse to near-sparse signals, and (ii) we provide a detailed mathematical and numerical justification of the usage of sparse recovery algorithms (including a bound on the reconstruction error) for this source model. Finally, extensive computer simulations are used to compare the performance of the proposed QNC scenario with respect to other network transmission scenarios. Specifically, our focus is to study the distributed compression capabilities of the proposed QNC scenario in a lossless scenario. The study of robust transmission in lossy cases will be done in a future work.

Although the idea of using compressed sensing has been initially proposed in [22], its theoretical and practical possibilities have not been studied by providing a mathematical formulation. Additionally, we discuss on using compressed sensing in a network coding-based scenario, which involves quantization and is different from the work in [22].

As another contribution of our work, we discuss the satisfaction of RIP in a network coding scenario, which has not been addressed in other works. Specifically, in [25,28], the authors do not discuss explicit conditions for which compressed sensing encoding (and decoding) works properly[a]. In this work, we propose conditions for network coding coefficients which ensure a robust recovery of messages, by using restricted isometry property.

Finally, our QNC scenario is different from other proposed schemes, in the sense that we perform quantization to fulfill limited lossless communication between the nodes, as opposed to only using analog network coding. Specifically, we study the behavior of the so-called tail probability [32] in our QNC scheme and show that its behavior is similar to that which is observed in the classic (identically and independently distributed (i.i.d.) Gaussian measurement matrix [33,34]. This leads us to conclude that our scheme requires a number of received measurements of the same order as that classic case (see Section 4). A detailed description of the data gathering scenario studied in this paper, as well as some notations, is presented in Section 2. In Section 3, we introduce and formulate our proposed Quantized Network Coding algorithm, followed by a discussion on its theoretical feasibility, using the restricted isometry property, in Section 4. In Section 5, we present the decoding algorithm used to recover quantized network coded packets and derive a performance bound on recovery error. Our simulation setup and results are presented in Section 6. Finally in Section 7, we conclude the paper with a discussion on the proposed method and ongoing work.

2 Problem description and notation

In this paper, we limit our study to a network with lossless links with limited capacity. This model could also correspond to lossy networks, where appropriate channel coding would have been applied. A more realistic lossy network model is left as a future work.

2.1 Network

As shown in Figure 1, we represent the network by a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes (vertices) and directed edges (links). Each node, v, is from the finite sorted set $\mathcal{V} = \{1, \ldots, n\}$, and each edge, e, is from the finite sorted set $\mathcal{E} = \{1, \ldots, |\mathcal{E}|\}$. Further, each edge (link) can maintain a lossless transmission from tail(e) to head(e), at a maximum finite rate of C_e bits per use. Transmission over each link is assumed to have no interference involved from other links or nodes. One may modify the capacities of each link to reflect the effect of interference over each link.

We define the sets of incoming and outgoing edges of node v, denoted by In(v) and Out(v), respectively, as follows:

$$\text{In}(v) = \{e : e \in \mathcal{E}, \text{ head}(e) = v\}, \quad (1)$$

$$\text{Out}(v) = \{e : e \in \mathcal{E}, \text{ tail}(e) = v\}. \quad (2)$$

The content of edge e at time instant t are represented by $Y_e(t)$, where t represents the discrete (integer) time index, during which a block of L channel symbols[b] is transmitted. $Y_e(t)$ is from a finite alphabet of size $\lfloor 2^{LC_e} \rfloor$, where $\lfloor . \rfloor$ denotes rounding down to the nearest integer. In the rest of the paper, the realizations of all capital letter random variables are denoted by lowercase letters.

2.2 Source signals

The nodes of the network are equipped with sensors; specifically, we model the sensed signals in each node v as

Figure 1 Directed graph representing a data gathering sensor network.
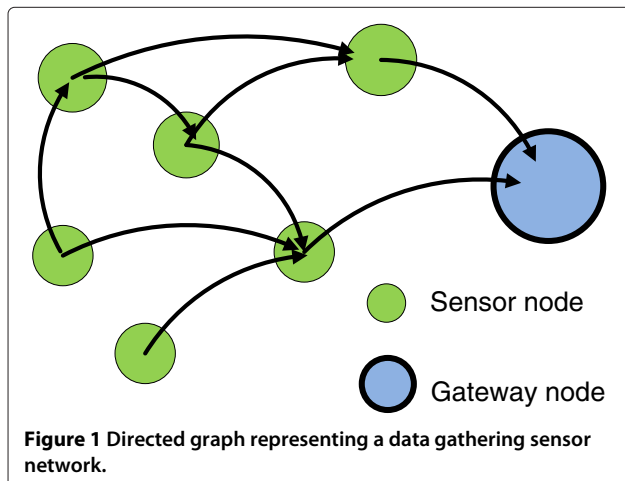
an information source, X_v, where $X_v \in \mathbb{R}$. To reflect the natural correlation between sensed data at each node, we assume that the set of signals X_v are near-sparse in some transform domain.

More specifically, defining the sorted vector[c] of X_v's,

$$\underline{X} = [X_v : v \in \mathcal{V}], \quad (3)$$

we assume that $\underline{X}$ is near-sparse in some orthonormal transform domain $\phi_{n \times n}$. Explicitly, for $\underline{S} = \phi^T \cdot \underline{X}$, and a small positive ϵ_k, we have

$$\frac{||\underline{S} - \underline{S}_k||_{\ell_1}}{||\underline{S}||_{\ell_1}} \leq \epsilon_k, \quad (4)$$

where $\underline{S}_k$ is such that

$$||\underline{S}_k||_{\ell_0} = k, \quad (5)$$

i.e., $\underline{S}_k$ is k-sparse. An example of the sparsifying transform matrix, ϕ, is the Karhunen Loeve transform of the messages.

Moreover, we assume that messages, X_v's, take their values in a bounded interval between $-q_{\max}$ and $+q_{\max}$. This is also a reasonable assumption as the sensing range of sensors is usually limited. The choice of $q_{\max}$ can be made after a statistical study of realizations of X_v's and can be chosen as some confidence region, in which most of the realizations of X_v's lay. Note that the sparsity model used in this paper is different from the conventional joint sparse model (JSM) [35], in that our node source signals or *messages* are scalar random variables, without correlation over time in each node. This is a valid assumption as a local transform coding could be applied to the time samples and generate a set of samples with no time redundancy.

2.3 Data gathering

Having these correlated information sources and the information network characterized, we study the transmission of X_v's to a single gateway node. The gateway or decoder node, denoted by v_0, $v_0 \in \mathcal{V}$, has high computational resources and is usually in charge of forwarding the information to a next level network, e.g., a wired backbone network. The described (single session) incast of sources to the unique decoder node is referred to as *data gathering*.

3 Quantized Network Coding

3.1 Principle

Random linear network coding for multicast of independent sources has been proposed and studied in [8], where the algebraic operations are carried out in a finite field. Since our work is motivated by the concepts of compressed sensing, in which the results are valid in the infinite field of real number, we have to use a real field alternative for conventional finite field network coding.

On the other hand, the finite capacity of the edges has to be appropriately coped with. As a result, we propose a method that we call *Quantized Network Coding*, which uses quantization to match infinite alphabet of real field network coded packets to the limited capacity of the network links.

In [31], for each network node $v \in \mathcal{V}$ and each outgoing edge $e \in \text{Out}(v)$, we defined QNC at node v, according to

$$Y_e(t) = \mathbf{Q}_e \left(\sum_{e' \in \text{In}(v)} \beta_{e,e'}(t) \cdot Y_{e'}(t-1) + \alpha_{e,v}(t) \cdot X_v \right),$$
(6)

for $t > 1$ and with $Y_e(1) = 0$, $\forall e \in \mathcal{E}$, to ensure initial rest condition in the network. This means that, at time t, the message on any outgoing edge of a node is made up of a quantized linear combination of the messages received by the node at the previous time instant and the information X_v measured by the node. The messages, X_v's, are assumed to be constant until the transmission is complete, which is why X_v's do not depend on t. The local network coding coefficients, $\beta_{e,e'}(t)$'s and $\alpha_{e,v}(t)$'s, are real-valued, and the determination of their value will be discussed in Section 4. The quantizer operator, $\mathbf{Q}_e(.)$, corresponding to outgoing edge e, is designed based on the values of C_e and L, and the distribution of its input (i.e., random linear combinations). A simple diagram of QNC at node v is shown in Figure 2.
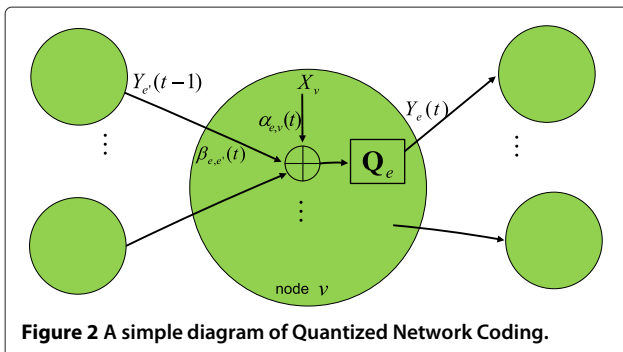
3.2 End-to-end equations
Denoting the quantization noise of $\mathbf{Q}_e(.)$ at time t, by $N_e(t)$, we can reformulate (6) as follows:

$$Y_e(t) = \sum_{e' \in \text{In}(v)} \beta_{e,e'}(t) \cdot Y_{e'}(t-1) + \alpha_{e,v}(t) \cdot X_v + N_e(t). \quad (7)$$

We define the adjacency matrix, $[F(t)]_{|\mathcal{E}| \times |\mathcal{E}|}$, as well as matrix $[A(t)]_{|\mathcal{E}| \times n}$ as

$$\{F(t)\}_{e,e'} = \begin{cases} \beta_{e,e'}(t), & \text{tail}(e) = \text{head}(e') \\ 0, & \text{otherwise} \end{cases}, \quad (8)$$

Figure 2 A simple diagram of Quantized Network Coding.

$$\{A(t)\}_{e,v} = \begin{cases} \alpha_{e,v}(t), & \text{tail}(e) = v \\ 0, & \text{otherwise} \end{cases}. \quad (9)$$

We also define the vectors of edge contents, $\underline{Y}(t)$, and quantization noises, $\underline{N}(t)$, according to

$$\underline{Y}(t) = [Y_e(t) : e \in \mathcal{E}], \quad (10)$$

$$\underline{N}(t) = [N_e(t) : e \in \mathcal{E}]. \quad (11)$$

As a result, (7) can be re-written in the following form:

$$\underline{Y}(t) = F(t) \cdot \underline{Y}(t-1) + A(t) \cdot \underline{X} + \underline{N}(t). \quad (12)$$

Depending on the network deployment, matrix $[B]_{|\text{In}(v_0)| \times |\mathcal{E}|}$ defines the relation between the content of edges, $\underline{Y}(t)$, and the received packets at the decoder node v_0. Explicitly, we define the vector of received packets at time t at the decoder:

$$\underline{Z}(t) = [Y_e(t) : e \in \text{In}(v_0)] = B \cdot \underline{Y}(t), \quad (13)$$

where

$$\{B\}_{i,e} = \begin{cases} 1, & i \text{ corresponds to } e \in \text{In}(v_0) \\ 0, & \text{otherwise} \end{cases}. \quad (14)$$

By considering (12) as the difference equation, characterizing a linear system with $\underline{X}$ and $\underline{N}(t)$'s as its inputs, and $\underline{Z}(t)$ its output, and using the results in [36], one gets

$$\underline{Z}(t) = \Psi(t) \cdot \underline{X} + \underline{N}_{\text{eff}}(t), \quad (15)$$

where the *measurement* matrix, $\Psi(t)$, and the *effective noise* vector, $\underline{N}_{\text{eff}}(t)$, are calculated as follows:

$$\Psi(t) = B \cdot \sum_{t'=2}^{t} F_{\text{prod}}(t'+1; t) A(t'), \quad (16)$$

$$\underline{N}_{\text{eff}}(t) = B \cdot \sum_{t'=2}^{t} F_{\text{prod}}(t'+1; t) \underline{N}(t'). \quad (17)$$

In (16) and (17), $F_{\text{prod}}(.;.)$ is defined as

$$F_{\text{prod}}(t_1; t_2) = \begin{cases} F(t_2) \cdot F(t_2 - 1) \cdots F(t_1), & t_2 \geq t_1 \\ \mathbf{I}_{|\mathcal{E}| \times |\mathcal{E}|}, & \text{otherwise} \end{cases} \quad (18)$$

and $\mathbf{I}$ denotes the identity matrix.

By storing $\underline{Z}(t)$'s, at the decoder, we build up the *total measurement vector*, $\underline{Z}_{\text{tot}}(t)$, as follows:

$$\underline{Z}_{\text{tot}}(t) = \begin{bmatrix} \underline{Z}(2) \\ \vdots \\ \underline{Z}(t) \end{bmatrix}_{m \times 1}, \quad (19)$$

where $m = (t-1)|\text{In}(v_0)|$. Therefore, the following can be established:

$$\underline{Z}_{\text{tot}}(t) = \Psi_{\text{tot}}(t) \cdot \underline{X} + \underline{N}_{\text{eff,tot}}(t), \quad (20)$$

where the $m \times n$ *total measurement matrix*, $\Psi_{\text{tot}}(t)$, and the *total effective noise* vector, $\underline{N}_{\text{eff,tot}}(t)$, are the concatenation result of measurement matrices, $\Psi(t)$'s, and effective noise vectors, $\underline{N}_{\text{eff}}(t)$. Because of our assumption to start transmission from $t = 1$, measurements in $\underline{Z}(1)$ are not useful for decoding, and therefore

$$\Psi_{\text{tot}}(t) = \begin{bmatrix} \Psi(2) \\ \vdots \\ \Psi(t) \end{bmatrix}, \tag{21}$$

$$\underline{N}_{\text{eff,tot}}(t) = \begin{bmatrix} \underline{N}_{\text{eff}}(2) \\ \vdots \\ \underline{N}_{\text{eff}}(t) \end{bmatrix}. \tag{22}$$

In conventional linear network coding, the total number of measurements, m (see (19)), is at least equal to the number of data sources, n (the number of nodes in the network here). Typically, the total measurement matrix is of full column rank, and if there is no uncertainty involved because of measurement noise, we are able to uniquely find a solution. In this paper, we are interested in investigating the feasibility of robust recovery of $\underline{X}$, when fewer number of measurements are received at the decoder than the number of messages, i.e., $m < n$.

The characteristic Equation (20) describing the QNC scenario can be treated as a compressed sensing measurement equation. This gives us an opportunity to apply the results in the literature of compressed sensing and sparse recovery [20,37] to our QNC scenario with near-sparse messages. However, one needs to examine the required conditions which guarantee sparse recovery in the proposed QNC scenario. In the following, we discuss theoretical and practical feasibility of robust recovery with a compressed sensing perspective.

4 Restricted isometry property
One of the main advantages of the compressed sensing approach is that it relies on a simple model of correlation for the sources; if sparse reconstruction can be applied successfully to recover $\underline{X}$ from Equation 20 at a given time t, this is achieved without requiring the encoders (network nodes) to know much about the underlying signal correlation. This section discusses the design of the linear mixing coefficients $\alpha_{e,v}(t)$ and $\beta_{e,e'}(t)$ and the impact of this design on the ability to apply sparse reconstruction techniques at the sink node v_0 to approximately recover the n source signals $\underline{X}$ from m measurements $\underline{Z}(t)$ at a given time t, where $m \ll n$.

4.1 The restricted isometry property
One of the properties that is widely used to characterize appropriate measurement matrices in the compressed

sensing literature is the *restricted isometry property* (RIP) [33]. Roughly speaking, this property provides a measure of norm conservation under dimensionality reduction [34]. In compressed sensing, the RIP of the measurement matrix between the sparse domain and the measurement domain allows to draw strong conclusions about the possibility to recover the original data from a small set of measurements [33]. In our case, this means that the RIP should hold for the measurement matrix $\Theta_{\text{tot}}(t) = \Psi_{\text{tot}}(t)\phi$.

An $m \times n$ matrix $\Theta_{\text{tot}}(t)$ is said to satisfy RIP of order k with constant δ_k, if for all k-sparse vectors $\underline{s}_k \in \mathbb{R}^n$, we have

$$1 - \delta_k \le \frac{\left\|\Theta_{\text{tot}}(t)\underline{s}_k\right\|_{\ell_2}^2}{\left\|\underline{s}_k\right\|_{\ell_2}^2} \le 1 + \delta_k. \tag{23}$$

Random matrices with i.i.d. zero-mean Gaussian entries are known to be appropriate measurement matrices for compressed sensing. Explicitly, an $m \times n$ i.i.d Gaussian random matrix, denoted $\mathbf{G}$, with entries of variance $\frac{1}{m}$, satisfies RIP of order k and constant δ_k, with a probability exceeding $1 - e^{-\kappa_1 m}$, (called *overwhelming* probability) if $m > \kappa_2 k \log(\frac{n}{k})$, where κ_1 and κ_2 only depend on the value of δ_k (theorem 5.2 in [38]).

Using the results above, it can be understood that an $m \times n$ i.i.d Gaussian random matrix, $\mathbf{G}$, satisfies RIP of order k, with a high probability, when the order of number of measurements, m, is $k \log(n/k)$, formally writing:

$$m = O\left(k \log\left(\frac{n}{k}\right)\right), \tag{24}$$

which is smaller than the order of n, the size of the data [38].

4.2 QNC design for RIP
We now turn to the design of QNC coefficients in Equation 6 so that the overall design satisfies RIP with high probability. We assemble here several results from the literature and additional simulations to motivate the proposed design.

In [31,32], we proposed a design for local network coding coefficients, $\beta_{e,e'}(t)$'s and $\alpha_{e,v}(t)$'s, which results in an appropriate total measurement matrix, $\Psi_{\text{tot}}(t)$, in the compressed sensing framework.

Theorem 1 (Theorem 3.1 in [32]). *Consider a Quantized Network Coding scenario, in which the network coding coefficients, $\alpha_{e,v}(t)$ and $\beta_{e,e'}(t)$, are such that:*

- $\alpha_{e,v}(t) = 0, \ \forall t > 2.$
- $\alpha_{e,v}(2)$'s are independent zero-mean Gaussian random variables.
- $\beta_{e,e'}(t)$'s are deterministic.

For such a scenario, the entries of the resulting $\Psi_{tot}(t)$ are zero-mean Gaussian random variables. Further, the entries of different columns of $\Psi_{tot}(t)$ are mutually independent. ∎

It is also numerically shown in [32] that a locally orthogonal set of $\beta_{e,e'}(t)$'s is a better choice than non-orthogonal sets[d]. This choice of coefficients is defined, for each node v and for all $e, e' \in \text{Out}(v)$, as

$$\sum_{e'' \in \text{In}(v)} \beta_{e,e''}(t) \cdot \beta_{e',e''}(t) = 0, \ e \neq e',$$

$$\sum_{e' \in \text{In}(v)} \beta_{e,e'}^2(t) = \frac{1}{|\text{In}(v)|^2}. \tag{25}$$

In cases where the number of outgoing edges is greater than the number of incoming edges, i.e., $|\text{Out}(v)| > |\text{In}(v)|$, some of the outgoing edges are randomly removed (not used for transmission) to ensure that the generated $\beta_{e,e'}(t)$'s are locally orthogonal. Furthermore, the second equation in (25) is a coefficient normalization which has no specific impact at this stage of the analysis, but which will be important in the study of bounds on sparse recovery performance in Section 5. Heuristically, such choice of orthogonal set makes each outgoing packets (of each node) to be innovative.

In [32], we established the relation between the satisfaction of RIP and the so-called tail probability

$$\mathbf{p}_{\text{tail}}(\Psi_{tot}(t), \varepsilon) = \max_{\underline{x}'} \mathbf{P}\left(\left| \left|\left| \Psi_{tot}(t)\underline{x}' \right|\right|_{\ell_2}^2 - 1 \right| > \varepsilon \right),$$

$$\text{subject to } \left|\left| \underline{x}' \right|\right|_{\ell_2} = 1 \tag{26}$$

by proving the following theorem.

Theorem 2 (Theorem 4.1 in [32]). *Consider $\Psi_{tot}(t)$ with the tail probability, as defined in (26), and an orthonormal transform matrix ϕ. Then, $\Theta_{tot}(t) = \Psi_{tot}(t) \cdot \phi$ satisfies RIP of order k and constant δ_k, with a probability exceeding,*

$$1 - \binom{n}{k}\left(\frac{42}{\delta_k}\right)^k \mathbf{p}_{tail}\left(\Psi_{tot}(t), \varepsilon = \frac{\delta_k}{\sqrt{2}}\right). \ ∎ \tag{27}$$

In [32], we have derived a detailed expression of the tail probability (26). Our ultimate goal would be to use this expression to directly conclude that the number of necessary measurements m in the QNC scenario is of the same order as that of a well-known Gaussian measurement matrix, as defined above. However, the relationship between the network and QNC parameters on the one hand and the measurement matrix $\Psi_{tot}(t)$ on

the other hand is too complicated to easily draw conclusions (see Equations 8, 9, and 16). We therefore resort to the following reasoning: we first show through simulations that the tail probabilities for the QNC and Gaussian measurement matrices are of the same order; we then conclude to a similar behavior of QNC and Gaussian measurement matrices in terms of RIP satisfaction and thus in terms of the required number of measurements.

In Figure 3, we present the numerical values of tail probabilities (defined in (26)) for the QNC measurement matrix $\Psi_{tot}(t)$, $\mathbf{p}_{\text{tail}}(\Psi_{tot}(t), \varepsilon)$, using the local network coding coefficients proposed in Theorem 1 with $\beta_{e,e'}(t)$'s satisfying the locally orthogonal condition[e] of (25). These tail probabilities are compared with those of i.i.d. Gaussian matrices, $\mathbf{G}$, versus the number of measurements, m, in each case.
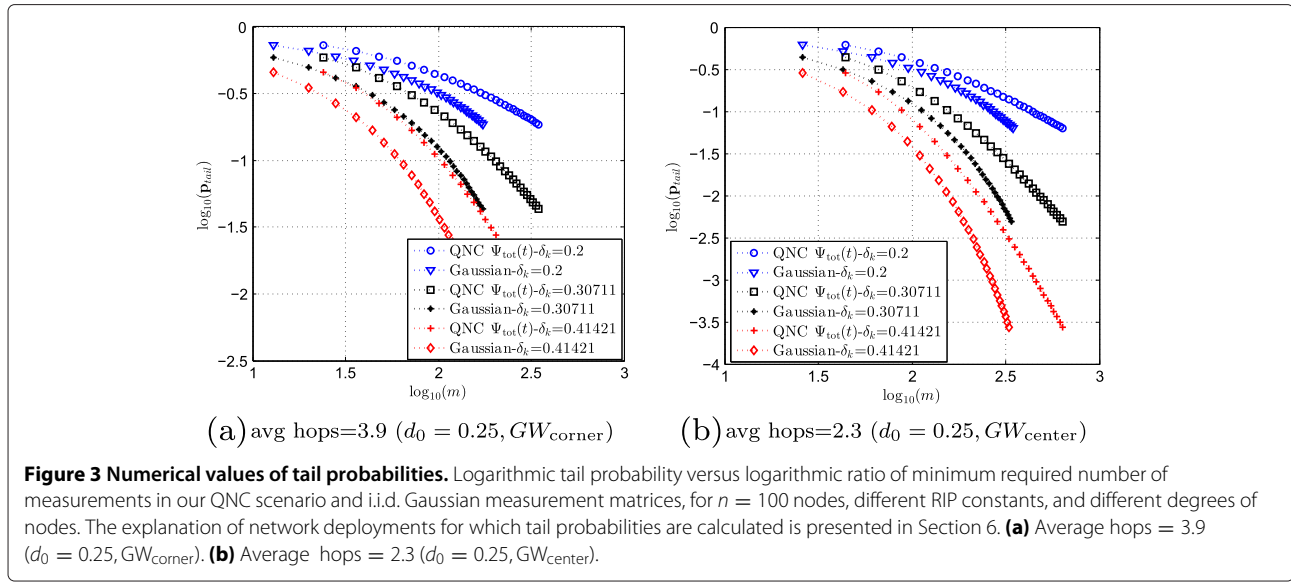
Our numerical evaluations in Figure 3 show that for the same value of tail probability, the QNC measurement matrix, $\Psi_{tot}(t)$, and the i.i.d. Gaussian matrix, $\mathbf{G}$, require a number of measurements m of the same order.

We can therefore also say, using Theorem 2, that the QNC measurement matrix, $\Psi_{tot}(t)$, and the i.i.d. Gaussian matrix, $\mathbf{G}$, have a similar behavior in terms of satisfying RIP as a function of m, so that they will typically require values of m of the same order to ensure sparse recovery.

In the following section, we extend our discussion to the robust recovery in QNC scenario, by using the guarantees, implied from the satisfaction of RIP.

5 Decoding using sparse recovery
In this section, we will explore the performance of decoding using sparse recovery based on Equation 20 and the QNC design proposed in Theorem 1. It is well known that recovery of exactly sparse vectors from an underdetermined set of linear measurements can be done with no error, using linear programming [39]. Specifically, theoretical works show that the NP-hard ℓ_0 minimization can be replaced with ℓ_1 minimization without any associated error, when dealing with noiseless measurements [37,39]. However, when dealing with noisy measurements, ℓ_1-min recovery does not necessarily offer a minimum mean squared error solution. There is still a lot of work being done to develop practical and near minimum mean squared error recovery algorithms for noisy cases. Sparse recovery from quantized measurements has been recently studied in a number of works [40-42]. For instance, the authors in [41] consider the estimation problem of sparse vectors from measurements that are quantized and corrupted by Gaussian noise. The main aspect that differentiates our model from that in [41] is that in our QNC scenario the resulting effective total measurement noises are non-linear functions of quantization noises at each edge.

Figure 3 Numerical values of tail probabilities. Logarithmic tail probability versus logarithmic ratio of minimum required number of measurements in our QNC scenario and i.i.d. Gaussian measurement matrices, for $n = 100$ nodes, different RIP constants, and different degrees of nodes. The explanation of network deployments for which tail probabilities are calculated is presented in Section 6. **(a)** Average hops = 3.9 $(d_0 = 0.25, \mathrm{GW}_{\mathrm{corner}})$. **(b)** Average hops = 2.3 $(d_0 = 0.25, \mathrm{GW}_{\mathrm{center}})$.

Along the lines of [20,33], the compressed sensing-based decoder for the QNC scenario solves the following convex optimization:

$$\hat{\underline{X}}(t) = \phi \cdot \arg \min_{\underline{S}'} \left|\left| \underline{S}' \right|\right|_{\ell_1},$$
$$\text{subject to } \left|\left| \underline{Z}_{\mathrm{tot}}(t) - \Psi_{\mathrm{tot}}(t)\, \phi\, \underline{S}' \right|\right|_{\ell_2} \leq \epsilon_{\mathrm{rec}}(t) \tag{28}$$

which can be solved by using linear programming [39]. The following theorems present our results on the recovery error using ℓ_1-min decoding of (28).

Theorem 3. *Consider the QNC scenario where the absolute value of messages are bounded by q_{max} and the local network coding coefficients are such that:*

- *$\alpha_{e,v}(t) = 0, \ \forall t > 2$.*
- *$\alpha_{e,v}(2)$'s are independent zero-mean Gaussian random variables with variance σ_0^2.*
- *$\beta_{e,e'}(t)$'s are deterministic and locally orthogonal according to (25).*

In such scenario, overflowing of linear combinations (over the limit of q_{max}) within the nodes happens with a probability less than or equal to

$$2|\mathcal{E}|Q(\sigma_0^{-1}), \tag{29}$$

where $Q(.)$ is the tail probability of standard normal distributions (i.e., one-sided Q function).

Proof. Using Cauchy-Schwartz inequality, for $t \geq 3$, we have

$$\sum_{e' \in \mathrm{In}(v)} |\beta_{e,e'}(t)| \leq \sqrt{\left(\sum_{e' \in \mathrm{In}(v)} |\beta_{e,e'}(t)|^2 \right) \left(\sum_{e' \in \mathrm{In}(v)} 1 \right)^2} \tag{30}$$

$$= \sqrt{\left(\frac{1}{|\mathrm{In}(v)|^2} \right) (|\mathrm{In}(v)|)^2} = 1 \tag{31}$$

As a result, since $\alpha_{e,v}(t)$'s are zero for $t \geq 3$, it is straightforward to imply that overflow may not happen for $t \geq 3$.

For $t = 2$, since only the node message X_v is available at each node, the values of $\beta_{e,e'}(2)$'s do not affect anything. Hence, only the value of $\alpha_{e,v}(2)$ can result in overflow and therefore $|\alpha_{e,v}(2)|$ should be less than or equal to one to prevent overflow. Moreover, because of the Gaussian distribution of $\alpha_{e,v}(2)$'s, each $\alpha_{e,v}(2)$ may have an absolute value more than one, with a probability of $2Q(\sigma_0^{-1})$. Therefore, using the union bound, the probability that there is at least one $\alpha_{e,v}(2)$ with $|\alpha_{e,v}(2)| > 1$ is upper bounded by $2|\mathcal{E}|Q(\sigma_0^{-1})$. □

Theorem 4. *Consider a QNC scenario where, for all $v \in \mathcal{V}$, the network coding coefficients satisfy the conditions in Theorem 3, and for which, based on the discussion in Section 4, the measurement matrix $\Theta_{tot}(t) = \Psi_{tot}(t)\phi$ satisfies RIP of order $2k$ with constant $\delta_{2k} < \sqrt{2} - 1$. The edge quantizers, $Q_e(.)$'s, are assumed to be uniform[f] with the step size Δ_e. Then, with a probability exceeding*

$$1 - 2|\mathcal{E}|Q(\sigma_0^{-1}), \tag{32}$$

for the ℓ_1-min decoding of (28), we have

$$\left\lVert \underline{\hat{X}}(t) - \underline{X} \right\rVert_{\ell_2} \leq c_1 \epsilon_{rec} + \frac{c_2}{\sqrt{2}} \epsilon_k, \tag{33}$$

where $\epsilon_{rec}^2(t)$ is defined in (34),

$$\epsilon_{rec}^2(t) = \frac{1}{4} \sum_{t'=2}^{t} \sum_{e \in In(v_0)} \left(\sum_{t''=2}^{t'} \sum_{e'=1}^{|\mathcal{E}|} |\{F_{prod}(t''+1; t')\}_{e,e'}| \, \Delta_{e'} \right)^2, \tag{34}$$

and the matrix product $F_{prod}(\bullet, \bullet)$ in (34) is defined in (18). The constants c_1 and c_2 are also defined as follows:

$$c_1 = 4 \frac{\sqrt{1 + \delta_{2k}}}{1 - (1 + \sqrt{2})\delta_{2k}}, \tag{35}$$

$$c_2 = 2 \frac{1 - (1 - \sqrt{2})\delta_{2k}}{1 - (1 + \sqrt{2})\delta_{2k}}. \tag{36}$$

Proof. According to Theorem 3, the conditions on the local network coding coefficients ensures that overflow does not happen with a probability exceeding $1 - 2|\mathcal{E}|Q(\sigma_0^{-1})$. Further, since the network is lossless, the only associated measurement noise is resulting from the quantization noise at the edges. For each uniform quantizer $\mathbf{Q}_e(\bullet)$, $e \in \mathcal{E}$, we have

$$-\frac{\Delta_e}{2} \leq N_e(t) \leq +\frac{\Delta_e}{2}. \tag{37}$$

This implies

$$\lVert \underline{N}_{eff,tot}(t) \rVert_{\ell_2}^2 = \sum_{t'=2}^{t} \sum_{i'=1}^{|In(v_0)|} \{\underline{N}_{eff}(t)\}_{i'}^2 \tag{38}$$

$$= \sum_{t'=2}^{t} \sum_{e \in In(v_0)} \left\{ \sum_{t''=2}^{t'} F_{prod}(t''+1; t') \underline{N}(t'') \right\}_e^2 \tag{39}$$

$$= \sum_{t'=2}^{t} \sum_{e \in In(v_0)} \left(\sum_{t''=2}^{t'} \sum_{e'=1}^{|\mathcal{E}|} \{F_{prod}(t''+1; t')\}_{e,e'} N_{e'}(t'') \right)^2 \tag{40}$$

$$\leq \sum_{t'=2}^{t} \sum_{e \in In(v_0)} \left(\sum_{t''=2}^{t'} \sum_{e'=1}^{|\mathcal{E}|} |\{F_{prod}(t''+1; t')\}_{e,e'}| \, |N_{e'}(t'')| \right)^2 \tag{41}$$

$$\leq \frac{1}{4} \sum_{t'=2}^{t} \sum_{e \in In(v_0)} \left(\sum_{t''=2}^{t'} \sum_{e'=1}^{|\mathcal{E}|} |\{F_{prod}(t''+1; t')\}_{e,e'}| \, \Delta_{e'} \right)^2 \tag{42}$$

$$= \epsilon_{rec}^2(t), \tag{43}$$

where (39) holds because of the one-to-one mapping structure of B matrix. This provides an upper bound on the ℓ_2-norm of measurement noise in our QNC scenario.

According to theorem 4.2 in [21], when the measurement matrix satisfies RIP of appropriate order and constant (as in the assumptions of Theorem 4) and the measurement noise is bounded, ℓ_1-min recovery can yield an estimate with bounded recovery error. Explicitly, the

bound is as in (33), considering the near-sparsity model of the messages and the obtained bound on the measurement noise. □

According to the preceding theorem, the upper bound, $c_1 \epsilon_{rec}$, is decreased when the quantization steps, Δ_e's, are decreased. Since $\Delta_e = 2q_{max}/2^{\lfloor LC_e \rfloor}$, a smaller upper bound on the ℓ_2 norm of the recovery error can be obtained by increasing the block length, L. Although this can be done practically, it will simultaneously increase the point to point transmission delays in the network, which may not be desirable. This creates a trade-off between reconstruction quality and delay, which will be explored in detail in Section 6.

As discussed in Theorem 4, the local network coding coefficients, proposed in (25), ensure that the normalization is respected and overflow does not happen, with high probability. More precisely, an appropriate choice of σ_0 should also be picked for this purpose. For example, when the number of edges is in the order of $1,000$, selecting $\sigma_0 = 0.25$ would result in a low probability for overflow.

It was also discussed in Section 4 that the resulting $\Theta_{tot}(t) = \Psi_{tot}(t)\phi$ satisfies the RIP condition with a high probability, when the local network coding coefficients are generated according to the assumptions of Theorem 1, with a number of measurements m of the same order as would be required for a i.i.d. Gaussian measurement matrix. Based on Theorem 4, if the resulting $\Psi_{tot}(t)$ satisfies the RIP of appropriate order with a high probability, then the robust recovery can be guaranteed with high probability.

Therefore, putting all these numerical and theoretical results together, QNC will result in bounded error recovery (33) with a number of measurements (number of packets received at the decoder) of smaller order than the number of messages. This saving in the required number of received packets can be interpreted as an *embedded distributed compression*, achieved by Quantized Network Coding at the nodes: the more packets are received at the decoder, the larger m will be and the lower the reconstruction error will be.

6 Simulation results

In this section, we evaluate the performance of Quantized Network Coding, by using different numerical simulations. The main motivation behind the proposed Quantized Network Coding technique is to allow for reconstruction of the correlated source signals at the sink node or decoder with a limited number of measurements. To this end, we will compare delay distortion curves for different data gathering algorithms. Our performance analysis includes statistical evaluation of the proposed QNC scenario versus packet forwarding and conventional finite field network coding schemes. The resulting analysis

will provide a comprehensive comparison between these transmission methods for different network deployments and correlation of sources.

6.1 Network deployment and message generation

To set up the simulations, we generate random deployments of networks with directed links, obtained from a transmission power loss model. Specifically, a certain number of nodes, n, are deployed in a unit square two-dimensional region, according to a uniform distribution. One of the deployed nodes in the network is randomly picked to be the gateway node, v_0, in which the messages are decoded. In our simulations, we examine two different probability models to pick the gateway node. In the first model, denoted by GW_{corner}, the gateway node is uniformly picked from the nodes within the region in the corners of the unit square, as shown in Figure 4a. In the second model, denoted by GW_{center}, the gateway node is uniformly picked from the nodes within the region in the center of a unit square, as shown in Figure 4a.

The asymmetric connectivity (which is different from full duplex transmission over links) of two nodes is determined according to an exponential power decay model: if there is a distance between node i and node j, denoted $d_{i,j}$, then there is an edge (link) from i to j; if

$$d_{i,j} \leq d_0, \tag{44}$$

and

$$P_{i,j} \leq P_0, \tag{45}$$

where d_0 is a threshold which determines the communication range of sensor nodes, $P_{i,j}$ is a uniform random variable between 0 and 1, and P_0 ($0 < P_0 \leq 1$) tunes the average percentage of nodes in the communication

range of a sensor toward which there will be a link. We change the value of d_0 (and typically keep $P_0 = 0.9$) to generate networks with different number of edges and different maximum hop distances, as described later in this section. Different settings for generating network deployments, the resulting average degree of nodes $|In(v)|$, and the resulting average hop distances of nodes from the gateway node are presented in Table 1.

In our simulation, each communication link (edges) can maintain a lossless communication of 1 bit per use, i.e., $C_e = 1$, for all $e \in \mathcal{E}$. We also assume that there is no interference involved from transmission in other nodes which may have been achieved by using a time multiplexing strategy. A sample network deployment is shown in Figure 4b, where the arrows represent the directed links between the nodes.

To generate a realization of messages, $\underline{x}$, we first generate a k-sparse random vector, $\underline{s}_k$, whose non-zero components are uniformly distributed between $-\frac{1}{2}$ and $+\frac{1}{2}$. Then, a near-sparse vector, $\underline{s}$, is obtained such that elements of $(\underline{s} - \underline{s}_k)$ are drawn from independent zero-mean uniform random variables and

$$\frac{||\underline{s} - \underline{s}_k||_{\ell_1}}{||\underline{s}||_{\ell_1}} = \epsilon_k. \tag{46}$$

This is followed by generation of an orthonormal random matrix, ϕ, and calculating random messages: $\underline{x} = \phi \cdot \underline{s}$. To ensure that x_j's are bounded, they are normalized between $-q_{max}$ and $+q_{max}$ (x_j's are multiplied by a constant value). The value of q_{max} used for the simulations does not affect the simulation results, since we are using average SNR as a measure of decoding quality. We study the performance of different transmission scenarios by
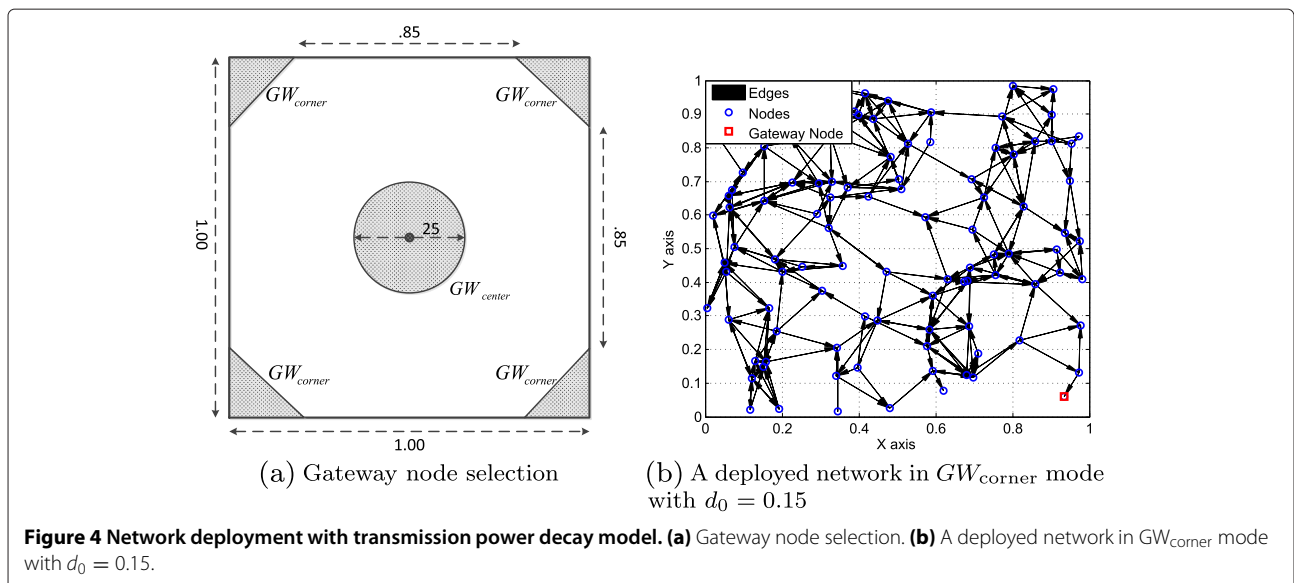
Figure 4 Network deployment with transmission power decay model. (a) Gateway node selection. **(b)** A deployed network in GW_{corner} mode with $d_0 = 0.15$.

Table 1 Number of hops in each deployment setting with $n = 100$ nodes

Simulation settings	Average $|\ln(v)|$	Average hops
$d_0 = 0.15, \text{GW}_{\text{corner}}$	5.3	9.7
$d_0 = 0.15, \text{GW}_{\text{center}}$	5.3	5.3
$d_0 = 0.25, \text{GW}_{\text{corner}}$	13.9	3.9
$d_0 = 0.25, \text{GW}_{\text{center}}$	13.9	2.3
$d_0 = 0.35, \text{GW}_{\text{corner}}$	24.8	2.7
$d_0 = 0.35, \text{GW}_{\text{center}}$	24.8	1.7

repeating our simulations for different values of sparsity factor, $\frac{k}{n}$, and near-sparsity parameter, ϵ_k.

The average signal-to-noise ratio (SNR) is used as the quality measure in our numerical comparisons. Explicitly, for the decoded messages in a scheme, $\hat{\underline{x}}$, the average SNR is defined as

$$\text{SNR} = 20 \log_{10} \frac{\overline{||\underline{x}||_{\ell_2}}}{\overline{||\underline{x} - \hat{\underline{x}}||_{\ell_2}}}, \qquad (47)$$

where $\overline{(.)}$ stands for the average over different realizations of network deployments. For each realization of network deployment, we only generate one realization of messages, and therefore, taking the average over different network deployments is enough to obtain the average SNR values.

The payback measure in our comparisons is the corresponding average delivery delay, to achieve the required quality of service (average SNR). Explicitly, delivery delay for a transmission which has terminated at t is equal to $(t - 1) L$ in all cases of transmission scenarios. In the case of packet forwarding, we do not consider the learning period required to find the routes from each sensor node to the decoder node.

The used simulation parameters are listed in Table 2. In the table, we describe the different simulated transmission scenarios.

6.2 Quantized Network Coding
For each generated random network deployment, we perform QNC with ℓ_1-min decoding. Local network coding

Table 2 The parameters of messages and the networks used in our simulations

Parameter	Value(s)
n	100
d_0	$0.15, 0.25, 0.35$
P_0	0.9
L	$1, \ldots, 40$
k/n	$0.01, 0.05, 0.10$
ϵ_k	$0, 0.002, 0.02, 0.2$

coefficients, $\alpha_{e,v}(t)$'s and $\beta_{e,e'}(t)$'s, are generated according to the conditions of Theorem 3, where $\sigma_0 = 0.25$. Edge quantizers, $\mathbf{Q}_e(.)$'s, have uniform characteristic with a range of $[-q_{\max}, +q_{\max}]$ and 2^L intervals (since $C_e = 1, \forall e$). Random $\alpha_{e,v}(2)$'s and $\beta_{e,e'}(t)$'s can be generated in a pseudo-random way, and therefore, only the generator seed needs to be transmitted to the decoder in a packet header.

At the decoder, the received measurements up to t, $\underline{z}_{\text{tot}}(t)$, are used to recover the original messages. Specifically, for a realization of messages, $\underline{x}$, we define $\hat{\underline{x}}_{\text{QNC}}(t)$ to be the recovered messages, using ℓ_1-min decoding, according to (28). The convex optimization, involved in (28), is solved by using the open source implementation of disciplined convex programming [43]. Moreover, the network deployment is assumed to be known at the decoder in order to build up $\Psi_{\text{tot}}(t)$ matrices (the random generator seed is enough to regenerate local network coding coefficients). Although the exact sparsity of messages, k, does not need to be known for performing ℓ_1-min decoding, the sparsifying transform, ϕ, should be known. The block length, L, has to be known at the decoder to be able to calculate the level of the effective measurement noise, i.e., $\epsilon_{\text{rec}}(t)$'s.

6.3 Quantization and packet forwarding
For each deployment, we also simulated a routing-based packet forwarding and compared it with the results for QNC. To find the routes from each node to the gateway node, we find the shortest path from each node to the gateway node using the Dijkstra algorithm [44]. Further, the real-valued messages, x_v's, are quantized at their corresponding source nodes, by using similar uniform quantizers, as used in QNC transmission. The system delivers all x_v's to the decoder node over a certain period of time and keeps track of delivered messages over time, t, in the recovered vector of messages, $\hat{\underline{x}}_{\text{PF}}(t)$. Moreover, if a message, x_v, is not delivered by time index, t, zero is used as its recovered value:

$$\{\hat{\underline{x}}_{\text{PF}}(t)\}_v = 0. \qquad (48)$$

6.4 Quantization and packet forwarding with CS decoding
The quantization and packet forwarding with CS decoding (QandPFwithCS) scenario is exactly the same as the quantization and packet forwarding (QandPF) scenario, except at the decoder side. Specifically, at the decoder node, if the messages of some nodes are still not delivered, the decoder tries to recover them from the other received (quantized) messages, using compressed sensing decoding. Explicitly, we define $\Psi_{\text{tot,PF}}(t)$ to be the mapping matrix from the messages to the received quantized messages, i.e.,

$$\{\Psi_{\text{tot.PF}}(t)\}_{i,\nu} = \begin{cases} 1, & \mathbf{Q}(x_\nu) \text{ delivered by } t \text{ and corresponds to } i\text{th received packet,} \\ 0, & \mathbf{Q}(x_\nu) \text{ not delivered by } t \end{cases}, \qquad (49)$$

and $\underline{z}_{\text{tot,PF}}(t)$ to be the set of received (delivered via PF) quantized messages at the decoder. In such case, the following ℓ_1 minimization is solved:

$$\hat{\underline{x}}_{\text{PFCS,0}}(t) = \phi \cdot \arg\min_{\underline{s}'} \left\| \underline{s}' \right\|_{\ell_1},$$

$$\text{subject to } \left\| \underline{z}_{\text{tot,PF}}(t) - \Psi_{\text{tot,PF}}(t)\phi\underline{s}' \right\|_{\ell_2} \qquad (50)$$

$$\leq \epsilon_{\text{rec,PF}}(t),$$

where $\epsilon_{\text{rec,PF}}(t)$ is the upper bound on the ℓ_2-norm of quantized delivered messages[g]. Then, for each ν if its quantized messages $\mathbf{Q}(x_\nu)$ is still not delivered, we use $\{\hat{\underline{x}}_{\text{PFCS,0}}(t)\}_\nu$, meaning

$$\{\hat{\underline{x}}_{\text{PFCS}}\}_\nu = \begin{cases} \mathbf{Q}(x_\nu), & \mathbf{Q}(x_\nu) \text{ delivered by } t, \\ \{\hat{\underline{x}}_{\text{PFCS,0}}(t)\}_\nu, & \mathbf{Q}(x_\nu) \text{ not delivered by } t \end{cases}.$$
$$(51)$$

As it can be predicted, compressed sensing-based decoding tries to find an approximate estimation for the un-delivered messages by using the redundancy of messages and improves the overall performance in terms of recovery error norm.

6.5 Quantization and network coding

Conventional finite field network coding is also simulated for transmission of messages to the decoder node. In this scenario, similar to packet forwarding, the messages are first quantized at their source nodes, by using a uniform quantizer. The quantizers have a range between $-q_{\max}$ and $+q_{\max}$, and their step size depends on the transmission block length, L. Then, the quantized messages are transmitted to the decoder node by running a classical batch-based finite field network coding [7,8]. The field size in network coding is determined by the value of L, and the network coding coefficients are picked randomly and uniformly from the field elements. At the decoder node, the received finite field packets are collected until n of them are stored, and the transmission is then stopped. If the finite field matrix, which maps the messages to the received packets at the decoder node, has full column rank, then the quantized messages can be reconstructed without any error. However, if the field size is not large enough and matrix inversion is not possible, then none of the messages can be decoded. In such case, we set the reconstructed (decoded) messages to be equal to their mean value (i.e., 0 in our simulations):

$$\{\hat{\underline{x}}_{\text{QandNC}}(t)\}_\nu = 0, \forall \nu. \qquad (52)$$

This is referred to as *all or nothing decoding* in the conventional network coding literature. Similar to QNC scenario, the network deployment is assumed to be known at the decode node, and the mapping matrix (from messages to received packets) can be built up by only receiving the seed of pseudo-random generators.
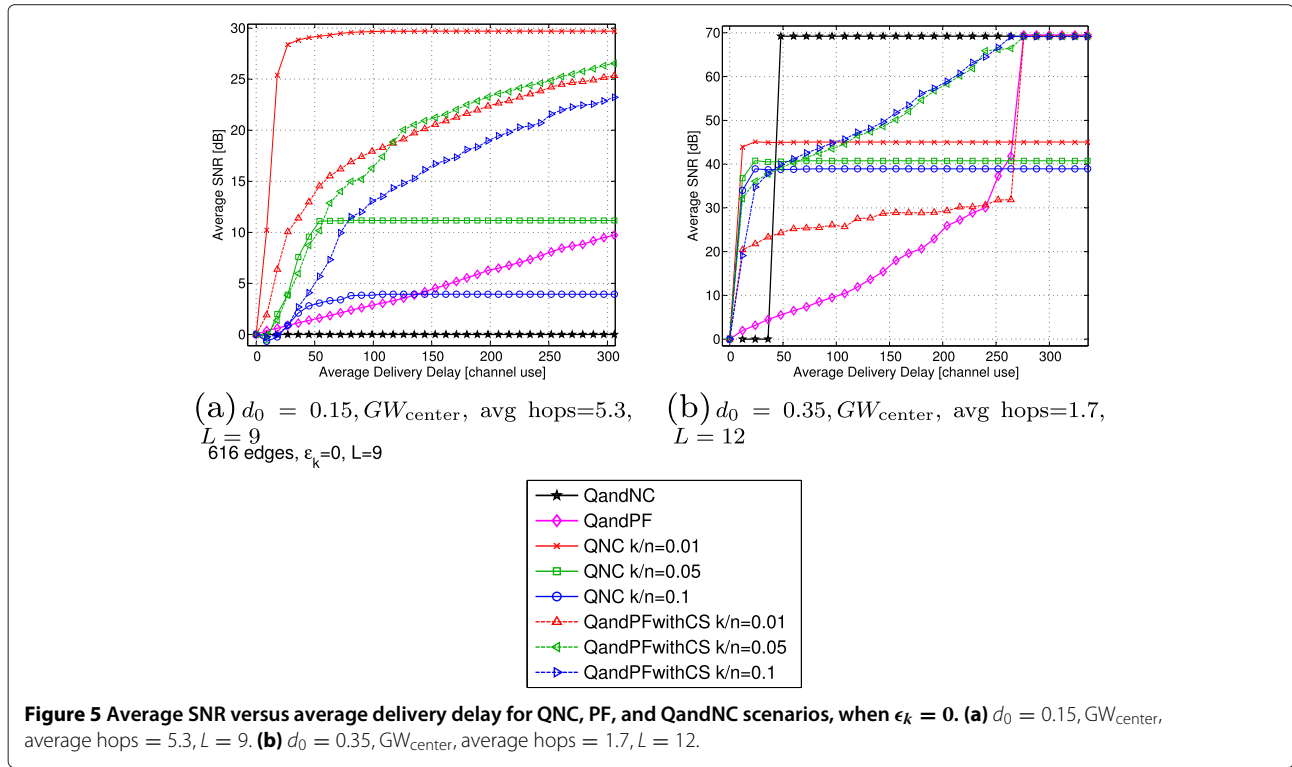
6.6 Analysis of simulation results

For a fixed block length, $L = 9$, the average SNR values versus the average delivery delay is depicted in Figure 5. In Figure 5a,b, the horizontal axis represents the product $(t - 1)L$, which is the delivery delay, corresponding to $L = 9$, for different values of $t \geq 1$. The vertical axis is the average SNR, calculated according to (47), for QNC, QandPF (with and without compressed sensing decoding), and quantization and network coding (QandNC) scenarios.

As it is shown in Figure 5a,b, when using the same block length, QNC achieves significant improvement, compared to PF, for low values of delivery delay. These low delays correspond to the initial t's in the transmission, at which a small number of packets are received at the decoder. As promised by the theory of compressed sensing, fewer measurements enable message recovery, with an associated measurement noise. After enough packets are received at the decoder, QNC achieves its best performance (where the curve is almost flat). This best performance improves (i.e., average SNR increases) when the correlation of messages is higher (sparsity factor k/n is lower).

The best performance for QandPF, QandPFwithCS, and QandNC happens after a longer period of time than for QNC. As it can be seen, this is the best achievable quality (SNR value), which is limited only by quantization noises at the source nodes, for both QandPF and QandNC scenarios. As it is also expected, using compressed sensing decoding (as in QandPFwithCS scenario) provides a better estimation of the messages before all the packets are delivered. Furthermore, as opposed to QandPF which shows a progressive improvement in the quality, QandNC has an all or nothing characteristic, as mentioned earlier. It is also interesting to note that low-density adjacency matrices in networks with small degree of nodes result in having (finite field) measurement matrices that are not of full rank in the QandNC scenario. Hence, as shown in Figure 5a, QandNC scheme fails to work properly.

The quantization noises and their propagation through the network does not allow QNC to achieve the same best performance as in PF and QandNC scenarios (where only

Figure 5 Average SNR versus average delivery delay for QNC, PF, and QandNC scenarios, when $\epsilon_k = 0$. (a) $d_0 = 0.15$, GW_{center}, average hops $= 5.3$, $L = 9$. **(b)** $d_0 = 0.35$, GW_{center}, average hops $= 1.7$, $L = 12$.

source quantization noise is involved). However, as it is shown in the following, QNC outperforms QandPF (with and without compressed sensing decoding) and QandNC scenarios in a wide range of delay values, when an appropriate block length is chosen.

After simulating QNC, QandPF, QandPFwithCS, and QandNC scenarios for different block lengths and calculating the corresponding delay and recovery error norms, we find the best values of block length for each specific average SNR value. The resulting L-optimized curves for each of these scenario are shown in Figure 6.

It can be seen in Figure 6a,b,c,d that, when the network does not have too many links (i.e., when the average hop distances are low), the proposed QNC scenario outperforms both routing-based packet forwarding (with and without compressed sensing decoding) and conventional QandNC scenarios. This is true for a wide range of average SNR values, varying up to around 35 dB, which is considered as high quality in many applications. Moreover, as it is expected, the average SNR of QNC scenario increases when the correlation of messages increases (i.e., when the sparsity factor, k/n, decreases).

As shown in Figure 6e,f, when dealing with networks with very high number of edges, which results in small average hop distances, the proposed QNC scenario cannot outperform QandNC scenario, for very high SNR values (explicitly for average SNR values higher than 40 dB). This may be a result of quantization noise pro-

pagation through the network during the QNC steps, which strengthen the effective measurement noise above the level that sparse recovery can compensate.

By comparing the figures, in which only the location of gateway node has changed, i.e., from GW_{center} to GW_{corner} (Figure 6a to Figure 6b and Figure 6c to Figure 6d), we can understand that QNC shows a more robust behavior than PF and QandNC schemes. In other words, QNC does not suffer from the complications (especially happening in packet forwarding) caused by asymmetric distribution of network flow. Using compressed sensing decoding for packet forwarding, as in QandPFwithCS scenario, improves the performance of packet forwarding in this situation, although it cannot outperform QNC scenario.

We have also studied the effect of the near-sparsity parameter, ϵ_k, on the performance of our QNC scheme. Those results are shown in Figure 7, where the average SNR is depicted versus the average delivery delay, for different settings of network deployment and a fixed sparsity factor of $k/n = 0.01$. Increasing the near-sparsity parameter, ϵ_k, means that the generated messages are getting further away from the sparsity model. As a result, the performance of QNC degrades when ϵ_k increases, which can be seen in Figure 7a,b,c,d,e,f. A more sophisticated correlation model, which would incorporate in the decoding procedure other prior information about the messages than only sparsity may improve the performance of the
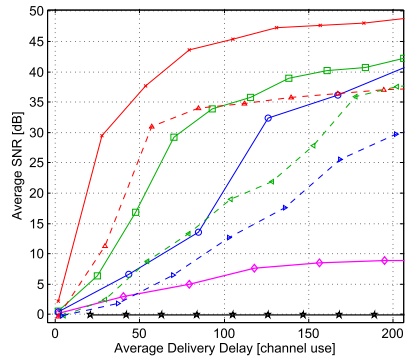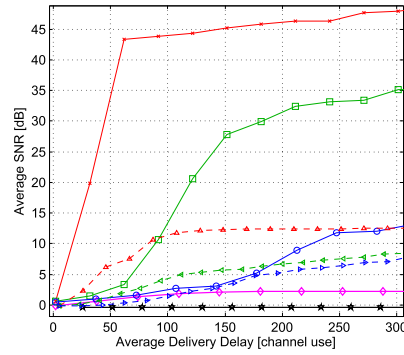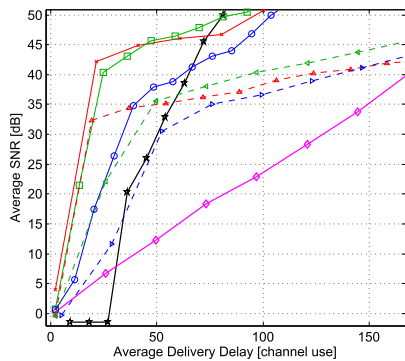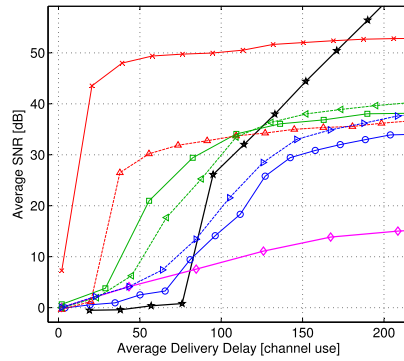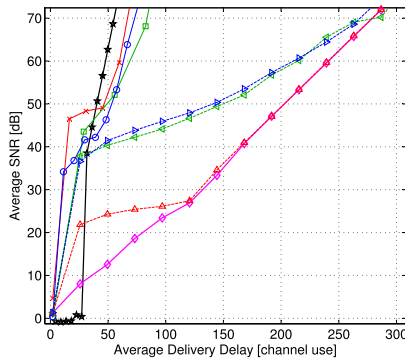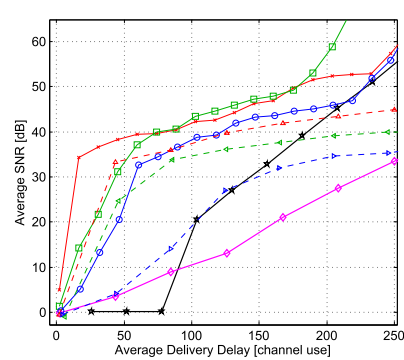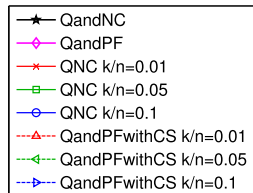
Figure 6 Average SNR versus average delivery delay for $\epsilon_k = 0$. (a) $d_0 = 0.15$, GW$_{\text{center}}$, average degree = 5.3, average hops = 5.3. **(b)** $d_0 = 0.15$, GW$_{\text{corner}}$, average degree = 5.3, average hops = 9.7. **(c)** $d_0 = 0.25$, GW$_{\text{center}}$, average degree = 13.9, average hops = 2.3. **(d)** $d_0 = 0.25$, GW$_{\text{corner}}$, average degree = 1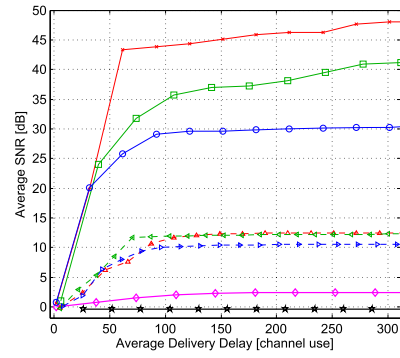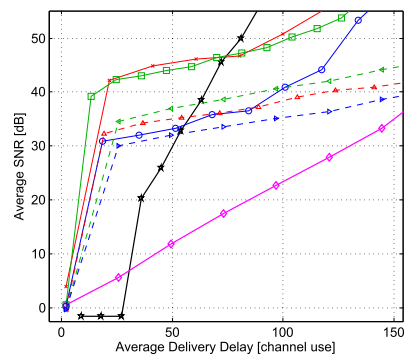3.9, average hops = 3.9. **(e)** $d_0 = 0.35$, GW$_{\text{center}}$, average degree = 24.8, average hops = 1.7. **(f)** $d_0 = 0.35$, GW$_{\text{corner}}$, average degree = 24.8, average hops = 2.7.
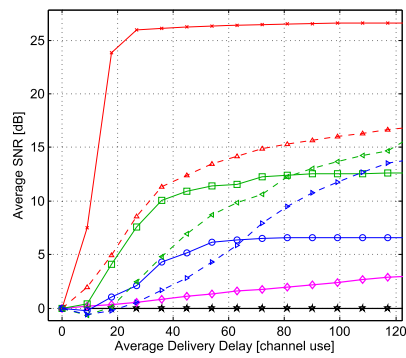
Figure 7 Average SNR versus average delivery delay for $k/n = 0.01$. (a) $d_0 = 0.15$, GW_{center}, average degree = 5.3, average hops = 5.3. **(b)** $d_0 = 0.15$, GW_{corner}, average degree = 5.3, average hops = 9.7. **(c)** $d_0 = 0.25$, GW_{center}, average degree = 13.9, average hops = 2.3. **(d)** $d_0 = 0.25$, GW_{corner}, average degree = 13.9, average hops = 3.9. **(e)** $d_0 = 0.35$, GW_{center}, average degree = 24.8, average hops = 1.7. **(f)** $d_0 = 0.35$, GW_{corner}, average degree = 24.8, average hops = 2.7.

QNC scenario. We are currently studying such possibility, and our initial findings are reported in [45,46].

In the routing-based packet forwarding scenarios (with and without compressed sensing decoding), the intermediate (sensor) nodes have to go through route training and queuing of packets. One of the main advantages of QNC is that the intermediate nodes should only carry out simple linear combination and quantization, which reduces the required computational power of intermediate sensor nodes (they still have to perform sensing and physical layer transmission). On the other hand, at the decoder sides, QNC requires an ℓ_1-min decoder which is potentially more complex than the receiver required for packet forwarding. However, since the gateway node is usually capable of handling higher computational operations, this may not be an issue in practical cases.
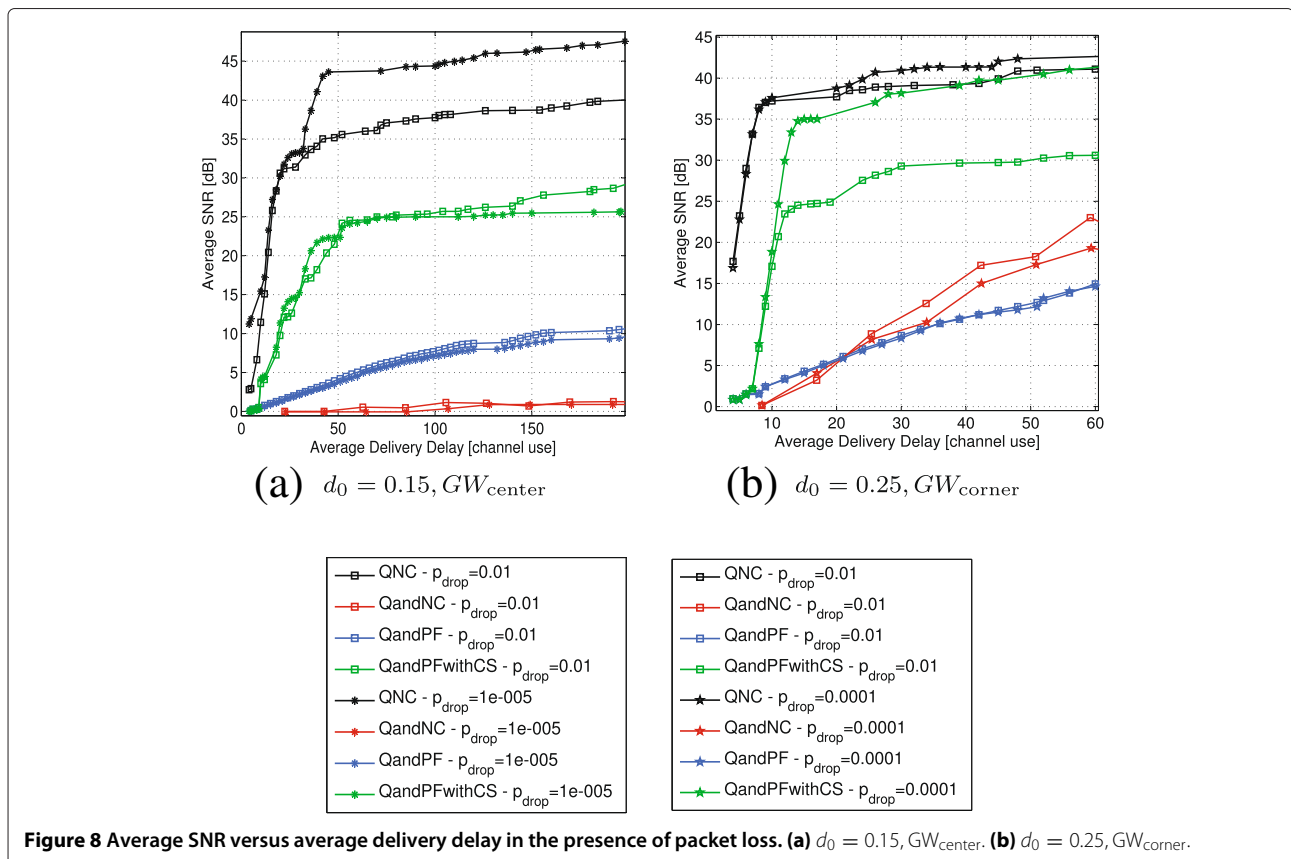
6.7 QNC in lossy networks

Although it is not the main focus of our paper, we have run some numerical simulations to assess the robustness of QNC scenario in lossy networks. Specifically, we consider a network model similar to the one used for the lossless case, but with the presence of packet losses. More precisely, all the links are assumed to have a bit dropping rate

of p_{drop}, i.e., a bit (which corresponds to a symbol in the case of $C_0 = 1$ considered in the simulations) is dropped (lost) during the transmission with a probability of p_{drop}. When dealing with packets of length L, a packet is considered as being dropped if one or more of its bits are lost. This will be applied to all different transmission schemes, described in Section 6.1.

During the packet forwarding, if a packet is not successfully transmitted over a channel, it needs to be retransmitted completely. Moreover, in the QandNC and QNC scenarios where finite field network coding and Quantized Network Coding are adopted, loss of a packet (transmitted over a link) is reflected by a zero value for the corresponding local network coding coefficient.

The simulation results for this lossy network scenario are shown in Figure 8. Similar to the case of lossless network, the curves are obtained by finding the appropriate packet length for each SNR value. We have used a wide range of bit loss rates p_{drop} for our simulations and shown results for a few representative values of p_{drop}. Specifically, we present the performance curves for a low loss rate of $p_{drop} = 10^{-5}, 10^{-4}$ and a high loss rate of $p_{drop} = 10^{-2}$.

Since the low SNR values (low decoding quality) in QNC scenario are obtained by using small packet lengths (small values of L), the probability of having a bit drop in

Figure 8 Average SNR versus average delivery delay in the presence of packet loss. (a) $d_0 = 0.15$, GW_{center}. **(b)** $d_0 = 0.25$, GW_{corner}.

the packet is smaller, compared to a larger packet length (larger L). As a result, the resulting performance curves are not very different, when having different loss rates. This is shown in Figure 8a,b, where there is a small gap between the curves of different p_{drop} values at low SNR values.

Moreover, since the compressed sensing decoder exploits the correlation between the messages, it is able to reconstruct some messages when their corresponding linear measurements are lost in the transmission. This fact can also be seen in QandPF scenario when compressed sensing decoding is adopted.

7   Conclusions

Joint source network coding of correlated sources was studied with a sparse recovery perspective. In order to achieve encoding of correlated sources without requiring the encoders to know the source correlation model, we proposed Quantized Network Coding, which incorporates real field network coding and quantization to take advantage of decoding using linear programming. Thanks to the work in the literature of compressed sensing, we discussed theoretical guarantees to ensure efficient encoding and robust decoding of messages. Moreover, we were able to make conclusive statements about the robust recovery of messages, when fewer number of received packets than the number of source signals (messages) were available at the decoder. Finally, our computer simulations verified the reduction in the average delivery delay, by using Quantized Network Coding.

Currently, we are studying the feasibility of near minimum mean squared error decoding, when other forms of prior information are available about the source. Specifically, we have suggested the use of belief propagation-based decoding [45] in a Bayesian scenario. However, more theoretical work is needed to derive mathematical guarantees for robust recovery. Studying the general case of lossy networks with interference between the links is also one of the proposed future directions.

Endnotes

[a] They only mention that dense networks satisfy restricted eigenvalue condition and do not prove it.

[b] Although the impact and value of L are not discussed at this point, it is an important design parameter, which will be extensively discussed in Section 6.

[c] In this paper, all the vectors are column-wise.

[d] This choice reduces the tail probabilities defined later on in Equation 26 and, as such, increases the probability of the measurement matrix satisfying RIP.

[e] Explicitly, we have a predetermined set of orthogonal matrices, used as $\beta_{e,e'}(t)$'s. Further, the variance of $\alpha_{e,v}(2)$'s are picked the same such that the mean of ℓ_2-norms (defined in [32]) is equal to 1.

[f] Although a uniform quantizer may not be the best choice for some message distributions, it is still widely used in practice. It also allows us to simplify the mathematical analysis to provide a theoretical bound on the resulting recovery error. The study of the impact of different quantizer designs is left as a future work.

[g] This depends on the characteristic of quantizers used at the source node to quantize each message before packet forwarding. Specifically, in our simulations where we used uniform quantizers with step size Δ_Q, $\epsilon_{\text{rec,PF}}(t)$ is equal to the product of Δ_Q and the number of delivered quantized messages.

References
1. I Akyildiz, W Su, Y Sankarasubramaniam, E Cayirci, A survey on sensor networks. IEEE Commun. Mag. **40**(8), 102–114 (2002)
2. C Chong, S Kumar, Sensor networks: evolution, opportunities, and challenges. Proc. IEEE. **91**(8), 1247–1256 (2003)
3. R Ahlswede, N Cai, S-Y Li, R Yeung, Network information flow. IEEE Trans. Inf. Theory. **46**, 1204–1216 (2000)
4. J Al-Karaki, A Kamal, Routing techniques in wireless sensor networks: a survey. IEEE Wireless Commun. **11**(6), 6–28 (2004)
5. T Ho, R Koetter, M Medard, M Effros, The benefits of coding over routing in a randomized setting, in *IEEE International Symposium on Information Theory,* 29 June–4 July 2003 (IEEE, Piscataway, 2003), p. 442
6. C Fragouli, Network coding for sensor networks, in *Handbook Array Processing Sensor Networks* (Wiley Online Library, 2009), pp. 645–667
7. R Koetter, M Médard, An algebraic approach to network coding. IEEE Trans. Netw. **11**(5), 782–795 (2003)
8. T Ho, M Medard, R Koetter, D Karger, M Effros, J Shi, B Leong, A random linear network coding approach to multicast. IEEE Trans. Inf. Theory. **52**, 4413–4430 (2006)
9. S Lim, Y Kim, A El Gamal, S Chung, Noisy network coding. IEEE Trans. Inf. Theory. **57**(5), 3132–3152 (2011)
10. A Dana, R Gowaikar, R Palanki, B Hassibi, M Effros, Capacity of wireless erasure networks. IEEE Trans. Inf. Theory. **52**, 789–804 (2006)
11. D Slepian, J Wolf, Noiseless coding of correlated information sources. IEEE Trans. Inf. Theory. **19**(4), 471–480 (1973)
12. Z Xiong, A Liveris, S Cheng, Distributed source coding for sensor networks. IEEE Signal Process. Mag. **21**(5), 80–94 (2004)
13. TS Han, Slepian-wolf-cover theorem for networks of channels. Inf. Control. **47**(1), 67–83 (1980)
14. T Ho, M Médard, M Effros, R Koetter, D Karger, Network coding for correlated sources, in *Proceedings of Conference on Information Sciences and Systems* (CiteSeer, 2004)
15. A Ramamoorthy, K Jain, PA Chou, M Effros, Separating distributed source coding from network coding. IEEE Trans. Netw. **14**, 2785–2795 (2006)
16. Y Wu, V Stankovic, Z Xiong, S Kung, On practical design for joint distributed source and network coding. IEEE Trans. Inf. Theory. **55**(4), 1709–1720 (2009)
17. G Maierbacher, J Barros, M Médard, Practical source-network decoding, in *6th International Symposium on Wireless Communication Systems* (IEEE, Piscataway, 2009), pp. 283–287
18. S Cruz, G Maierbacher, J Barros, Joint source-network coding for large-scale sensor networks, in *IEEE International Symposium on Information Theory Proceedings* (IEEE, Piscataway, 2011), pp. 420–424

19. F Kschischang, B Frey, H Loeliger, Factor graphs and the sum-product algorithm. IEEE Trans. Inf. Theory. **47**(2), 498–519 (2001)
20. D Donoho, Compressed sensing. IEEE Trans. Inf. Theory. **52**, 1289–1306 (2006)
21. R Baraniuk, M Davenport, M Duarte, C Hegde, *An Introduction to Compressive Sensing* (Addison-Wesley, Boston, 2011)
22. J Haupt, W Bajwa, M Rabbat, R Nowak, Compressed sensing for networked data. IEEE Signal Process. Mag. **25**, 92–101 (2008)
23. N Nguyen, D Jones, S Krishnamurthy, Netcompress: coupling network coding and compressed sensing for efficient data communication in wireless sensor networks, in *2010 IEEE Workshop on Signal Processing Systems,* Oct 2010 (IEEE, Piscataway, 2010), pp. 356–361
24. C Luo, F Wu, J Sun, CW Chen, Compressive data gathering for large-scale wireless sensor networks, in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking,* MobiCom '09, (ACM, New York, 2009), pp. 145–156
25. S Feizi, M Médard, M Effros, Compressive sensing over networks, in *48th Annual Allerton Conference on Communication, Control, and Computing* (IEEE, Piscataway, 2010), pp. 1129–1136
26. W Xu, E Mallada, A Tang, Compressive sensing over graphs, in *IEEE International Conference on Computer Communications (INFOCOM)* (IEEE, Piscataway, 2011), pp. 2087–2095
27. M Wang, W Xu, E Mallada, A Tang, Sparse recovery with graph constraints: fundamental limits and measurement construction, in *IEEE International Conference on Computer Communications (INFOCOM)* (IEEE, Piscataway, 2012), pp. 1871–1879
28. S Feizi, M Medard, A power efficient sensing/communication scheme: joint source-channel-network coding by using compressive sensing, in *49th Annual Allerton Conference on Communication, Control, and Computing* (IEEE, Piscataway, 2011), pp. 1048–1054
29. F Bassi, L Chao, L Iwaza, M Kieffer, Compressive linear network coding for efficient data collection in wireless sensor networks, in *Proceedings of the 2012 European Signal Processing Conference* (IEEE, Piscataway, 2012), pp. 1–5
30. B Dey, S Katti, S Jaggi, D Katabi, M Medard, S Shintre, "Real" and "complex" network codes: promises and challenges, in *Fourth Workshop on Network Coding, Theory and Applications. 2008 NetCod 2008* (IEEE, Piscataway, 2008), pp. 1–6
31. M Nabaee, F Labeau, Quantized network coding for sparse messages, in *IEEE Statistical Signal Processing Workshop,* Ann Arbor, Aug 2012 (IEEE, Piscataway, 2012), pp. 832–835
32. M Nabaee, F Labeau, Restricted isometry property in quantized network coding of sparse messages, in *IEEE Global Telecommunications Conference* Anaheim, Dec 2012 (IEEE, Piscataway, 2012)
33. EJ Candes, The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique. **346**(9-10), 589–592 (2008)
34. R Baraniuk, Compressive sensing. IEEE Signal Process. Mag. **24**(4), 118–121 (2007)
35. MF Duarte, S Sarvotham, MB Wakin, D Baron, RG Baraniuk, Joint sparsity models for distributed compressed sensing, in *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations* (IEEE, Piscataway, 2005)
36. T Kailath, *Linear Systems*, vol. 1. (Prentice-Hall, Englewood Cliffs, 1980)
37. E Candes, J Romberg, Sparsity and incoherence in compressive sampling. Inverse Probl. **23**(3), 969 (2007)
38. R Baraniuk, M Davenport, R Devore, M Wakin, A simple proof of the restricted isometry property for random matrices. Constr. Approx. **28**(3), 253–263 (2007)
39. E Candes, T Tao, Decoding by linear programming. IEEE Trans. Inf. Theory. **51**, 4203–4215 (2005)
40. W Dai, HV Pham, O Milenkovic, Distortion-rate functions for quantized compressive sensing, in *IEEE Information Theory Workshop on Networking and Information Theory* (IEEE, Piscataway, 2009), pp. 171–175
41. A Zymnis, S Boyd, E Candes, Compressed sensing with quantized measurements. IEEE Signal Process. Lett. **17**(2), 149–152 (2010)
42. L Jacques, DK Hammond, JM Fadili, Dequantizing compressed sensing: when oversampling and non-Gaussian constraints combine. IEEE Trans. Inf. Theory. **57**(1), 559–571 (2011)
43. M Grant, S Boyd, CVX: Matlab software for disciplined convex programming, version 1.21 (2012). http://cvxr.com/cvx. Accessed Aug 2012
44. E Dijkstra, A note on two problems in connexion with graphs. Numerische Mathematik. **1**(1), 269–271 (1959)
45. M Nabaee, F Labeau, Non-adaptive distributed compression in networks, in *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)* (IEEE, Piscataway, 2013), pp. 239–244
46. M Nabaee, F Labeau, Bayesian quantized network coding via generalized approximate message passing, in *2014 Wireless Telecommunications Symposium* Washington, DC Apr 2014 (IEEE, Piscataway, 2014)